IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2019, accepted February 25, 2019, date of publication March 8, 2019, date of current version April 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902429

Exploring Competitive Features Using Deep Convolutional Neural Network for Finger Vein Recognition

YU LU[1], SHANJUAN XIE[2], AND SHIQIAN WU[3], (Senior Member, IEEE)

[1]SenseTime Research, Shenzhen 518000, China
[2]College of Science, Institute of Remote Sensing and Earth Science, Hangzhou Normal University, Hangzhou 311121, China
[3]Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology, Wuhan 430081, China

Corresponding author: Shiqian Wu (shiqian.wu@wust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61775172, Grant 61701153, and Grant 61371190, and in part by the Open Project of the Key Laboratory of Electronics and Information Applied Technology of Crime Scene Investigation, Ministry of Public Security of China, under Grant EISI2016003.

ABSTRACT Deep convolutional neural networks (DCNN) have been applied successfully for finger vein recognition and have achieved promising performance in the past three years. However, public finger vein datasets are scarce and tend to be relatively small, and thus unable to provide a substantial number of well-labeled images needed to train an effective convolutional neural network (CNN). In this paper, a CNN model pre-trained on ImageNet is adopted to develop a CNN-based local descriptor named CNN competitive order (CNN-CO) that can exploit discriminative features for finger vein recognition. The CNN filters from the first layer of the AlexNet network and the corresponding CNN filtered images are visualized and compared with Gabor filters. According to the appearances and outputs of the CNN filters in three color spaces, we select the ones that most resemble the Gabor filters. The selected CNN filters are employed to generate a competitive order (CO) image using the winner-take-all rule. Then, the pyramidal histograms calculated from the CO image in different levels are concatenated to build the final histogram. The extensive experimental results on two public finger vein datasets demonstrate the effectiveness of the proposed method in selecting CNN filters. The results show that the proposed CNN-CO scheme using the selected CNN filters outperforms the well-known local descriptors.

INDEX TERMS Competitive pattern, finger vein, convolutional neural network.

I. INTRODUCTION

Due to the tremendous growth in demand for secure systems, automatic personal authentication using biometrics has attracted significant attention. Biometrics technology has reached a mature stage of development over the last two decades. As an emerging form of automated identification, finger vein biometrics has several important characteristics. It is a user-friendly technique that can be used with small imaging devices while providing high quality liveness, detection and high-level security [1]–[3]. In addition, the structure of the blood vessels located underneath the skin is unique for each individual and remains consistent over a long period of time. The property of the internal blood vessel network makes finger vein identification systems immune to forgery.

Deep convolutional neural networks (CNNs) have demonstrated excellent performance for image classification, as shown in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4] and in scene understanding [5]. Inspired by the remarkable success of CNNs in image classification, a number of CNN-based approaches have been proposed for tackling various problems in computer vision such as object detection [6], [7], object tracking [8], [9], segmentation [10], [11], and video classification [12], [13]. Generally, the success of CNN-based approaches depends on a large labeled dataset and the ability to train well-designed learning system using the labeled dataset. However, in finger vein recognition, public datasets are quite scarce and relatively small in comparison with the public datasets for face recognition and scene understanding. It is difficult to train an excellent CNN model to provide promising performance for finger vein recognition using a limited dataset.

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Wu.

An alternative approach is to exploit features directly from a trained CNN model. Gong *et al.* [14] introduced a framework using multi-scale orderless pooling (MOP) to aggregate CNN activations from high layers using a vector of locally aggregated descriptor (VLAD). In this framework, the deep activation features are extracted directly from a CNN model pre-trained on ImageNet. Wang *et al.* [15] proposed to adopt a CNN model pre-trained on ImageNet as a universal feature descriptor. This approach represents a hand vein image by concatenating the local features from different levels of the spatial pyramid. Inspired by the successful applications of a pre-trained CNN model, we propose to devise an approach that can combine a pre-trained CNN model with the characteristics of finger vein biometrics. However, the differences between images from ImageNet and finger vein images lie not only in the images themselves, but also in the image space. Consequently, developing a method for exploiting effective features using a CNN model pre-trained on ImageNet is not straightforward.

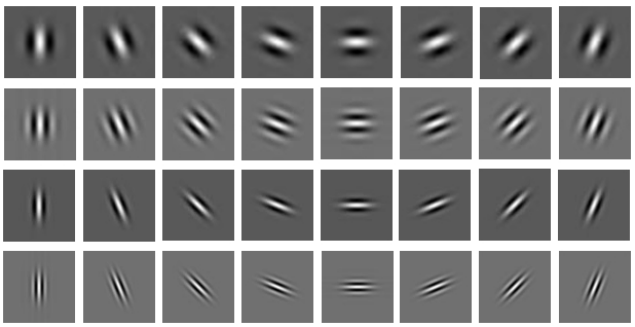

FIGURE 1. Gabor filters with four groups of parameters on eight orientations.

Finger veins located underneath the skin of a finger contain a vascular network that resembles a piecewise linear pattern, which are similar to the principle lines in a palm print. For biometrics involving abundant line structures, orientation coding (OC)-based approaches using the Radon transform [16], [17] and tunable filters such as steerable filter [18] and Gabor filter [3], [19]–[21], are appropriate for capturing the orientation information. OC-based representation is normally obtained by first convolving an image with a bank of filters and then encoding the filtered responses using several rules. Since OC-based approaches can achieve guaranteed accuracy, they have been applied widely for palmprint recognition and finger vein recognition. Figs. 1 and 2 depict the Gabor filters and their filtered images. In Fig. 3(a), the visualization of convolutional filters from the first layer of the AlexNet [4] network indicates that some CNN filters from the first layer and their outputs are similar to Gabor filters. This similarity motivates us to select CNN filters from a shallow layer to extract orientation features for finger vein recognition.

In this paper, we propose to employ the learnable convolutional filters from a CNN model pre-trained on ImageNet to extract the CNN competitive order (CNN-CO) for finger vein recognition. Specifically, the CNN filters from the first convolutional layer of the AlexNet network are employed as these filters with large kernel size well capture line structure features. By visualizing the CNN and Gabor filters and their outputs, five labels are given to the CNN filters according to their appearances and filtered images. Applying these labels, the CNN filters that can capture line structure are selected to establish a new group of CNN filters. Similar to the OC-based approach, the competitive order (CO) map can be computed using the winner-take-all rule on the new group of CNN filtered images. Then the pyramid histograms of the competitive order are extracted and concatenated to generate the final histogram for an image representation.

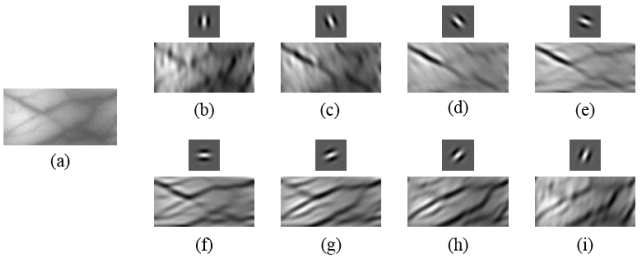

FIGURE 2. Gabor filtered images. (a) ROI image, (b-i) Gabor filters (the first and third rows), and the corresponding filtered images (the second and fourth rows). The orientations for (b-i) are 0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5. The pixels in these images are normalized to [0, 255]. For visualization, λ, ψ, γ, and σ are selected as 10, 0, 1, and 6.66, respectively.

This paper makes three important contributions to the field. (1) A framework that employs a CNN model pre-trained on ImageNet for finger vein recognition is presented. Different from the existing CNN-based technologies that require a large labeled dataset to train a deep model, the proposed method is more readily usable in fields where publicly available datasets are scarce and tend to be small, such as finger vein recognition. In addition, only the CNN filters from the first layers are used. (2) We define five labels to the CNN filters according to their appearances and outputs. The proposed method confirms that CNN filters with different labels show various degrees of effectiveness for feature representation. (3) The proposed CNN-CO can perform better than the state-of-the-art algorithms, as demonstrated in experimental work.

The remainder of this paper is organized as follows. Section 2 introduces the related work on finger vein recognition technology. A comparison of the visualization of the Gabor filters and CNN filters is presented in Section 3. Section 4 describes our proposed approach, while Section 5 provides extensive experimental results and analysis. Finally, Section 6 provides our conclusion and considerations for future work.

II. RELATED WORK

The last decade has witnessed extensive development of automated finger vein recognition technology. Like other biometric techniques, finger vein recognition consists of four main steps: image collection, image preprocessing, feature extraction, and matching. Many notable techniques have been proposed for finger vein recognition.

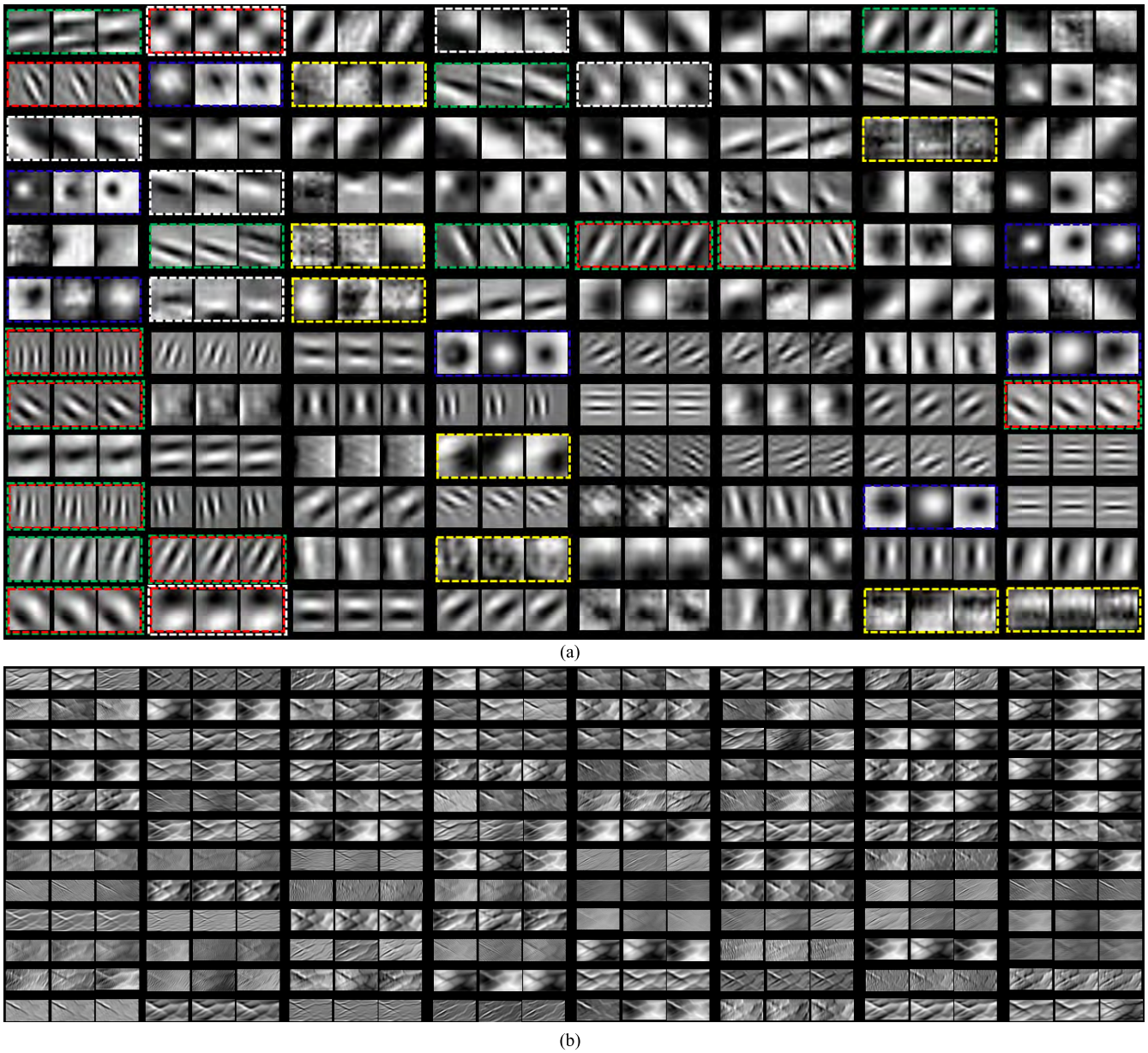

FIGURE 3. Visualizations of CNN filters and their outputs: (a) The 96 groups of CNN filters from the first layer of the AlexNet network; (b) CNN filtered images. The input image is Fig. 2(a). Filter (m, n) represents the image coordinate. For example, CNN filter (6, 1) describes the filter from the sixth row, first column. Green box: *line* filter, red box: *similar* filter, blue box: *circle* filter, white box: *smooth* filter, and yellow box: *noisy* filter.

Existing feature extraction approaches can be divided into several different categories, including methods based on vein pattern recognition, subspace learning, local descriptors, and CNNs. The pioneering pattern-based methods utilized the repeated line tracking [22] and the local maximum curvature [23]. To reduce the dependence on the parameters in these two methods, Song *et al.* [24] introduced a finger vein pattern extraction method using the mean curvature. However, segmentation errors could occur because of low quality finger vein images, and these errors could degrade the recognition performance. In the category of subspace learning methods, Wu and Liu [25] presented a finger vein identification system using principal component analysis (PCA) and neural network techniques. Wu and Liu [26] applied PCA and linear discriminant analysis (LDA) in image preprocessing for dimension reduction and feature extraction. Subspace learning based methods reshape 2-D image data into 1-D vectors. Consequently, the local features are ignored in these methods.

As a typical feature representation approach, local descriptor-based methods have been studied for finger vein recognition. Local binary pattern (LBP) [27], [28] and local line binary pattern (LLBP) [29] approaches were suggested for extracting local features from finger vein images. To explore more discriminative local features by utilizing the line structure of finger vein images, researchers have proposed OC-based methods employing edge operators [30] and Gabor filters [3] with competitive rules. The local

descriptor-based methods can achieve impressive matching accuracy.

Motivated by the success of CNNs for image classification, CNNs have been studied for finger vein recognition. DeepVein [31] was proposed for finger vein verification based on a deep CNN. This work fine-tuned the VGG-16 network for feature extraction. Hong *et al.* [32] also employed VGG-16 architecture for finger vein recognition. In response to the limitations of datasets available for training, Xie and Kumar [33] presented a light CNN with supervised discrete hashing for finger vein identification. To reduce dependence on the availability of a large image dataset for training, Fang *et al.* [1] introduced a novel finger vein verification system based on a two-stream convolution network learning approach. In addition to the adoption of CNNs for feature extraction, CNNs have also been used for finger vein image quality assessment [34].

In contrast with the approaches that train CNN models using the limited available finger vein image datasets, a CNN model pre-trained on ImageNet is adopted for feature extraction in this paper. The CNN filters similar to Gabor filters that capture line structures are selected and a local descriptor using the selected CNN filters is proposed for finger vein recognition. Consequently, the proposed approach does not need to develop a strategy for training a CNN model using the limited finger vein datasets.

III. VISULIZATION OF FILTERS AND FILTER OUTPUTS

Finger vein images are full of line segments in various orientations. The Gabor filter is a well-known handcrafted filter that captures discriminative features from different orientations and frequencies. In this section, we visualize the Gabor filters and CNN filters from the first layer of the AlexNet network and compare their filtered images.

A. GABOR FILTERS

Generally, the even-symmetric component of a Gabor filter, is used for extracting features from biometric images, as shown below:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp(-\frac{x_\theta^2 + \gamma^2 y_\theta^2}{2\sigma^2}) \cos(2\pi \frac{x_\theta}{\lambda} + \psi), \tag{1}$$

where $x_\theta = x\cos\theta + y\sin\theta$ and $y_\theta = -x\sin\theta + y\cos\theta$; λ denotes the wavelength of the sinusoidal factor; θ represents the orientation of normal to parallel stripes of a Gabor function; ψ is the phase offset, σ is the standard deviation of the Gaussian envelope; and γ is the spatial aspect ratio.

Assume that $I(x, y)$ is an input image. $F_k(x, y)$ denotes one of the corresponding Gabor filtered images at orientation index k, which can be expressed as:

$$F_k(x, y) = G_k(x, y) * I(x, y), \tag{2}$$

$$\theta = k\frac{\pi}{K}, \tag{3}$$

where $k = 1, 2, 3, \cdots, K$, and $*$ denotes the convolution operator.

Fig. 1 shows four groups of Gabor filters, in which each group contains eight orientations and varies in frequency. As illustrated in Fig.1, the handcrafted filters are smooth and display line structures with various orientations. Fig. 2 shows the filtered images obtained using the first group of Gabor filters from Fig. 1. The filtered images depict their ability to capture directional features from different orientations. The effectiveness of employing Gabor filters for capturing local directional features has been illustrated in [3], [20], [35] and [36].

B. CNN FILTERS

Unlike handcrafted filters which are designed without any learning process, deep CNN-based feature extraction is a data-driven technique that can learn robust representations from data. The performance of a deep CNN model and its generalization ability is based primarily on the use of an extensive image dataset with labels and deep models with handcrafted modules (e.g., pooling, batch normalization). With the availability of a huge quantity of image data and label information for training, the trained deep CNN generally performs better than the handcrafted filters in computer vision missions [4].

Compared with handcrafted filters, CNN filters show the superiority of their feature representation from various layers with different depths. Each convolution layer contains a group of learnable filters with a different number of channels. As introduced in [37], layers from a well-trained deep CNN show the hierarchical nature of features. For the AlexNet network, Layer 2 focuses on corners and other edges or color conjunctions. Layer 3 contains more complex invariances, capturing similar textures. Layer 4 shows significant variations and is more class-specific. Layer 5 describes entire objects that have significant pose variations.

In contrast with the deeper convolution layers, the first convolution layer in a deep CNN model is unique. The input of the first convolution layer is the raw pixel data, and filters from the first layer are a mix of extremely high and low frequency information [37]. Well-trained CNN networks usually contain smooth filters without noisy patterns. In addition, filters from the first layer of AlexNet network have a larger kernel size than filters from the deeper layers. A filter with a larger kernel size captures the line structure of finger veins well. Fig. 3(a) visualizes the 96 groups of CNN filters from the first layer of the AlexNet network. Both of the height and width of each CNN filter shown in Fig. 3(a) are 11. Since the AlexNet network is trained using color images, the RGB filters are included for comparison. Each group shown in Fig. 3(a) includes three CNN filters from RGB spaces.

It is observed from Fig. 3 that many groups of filters show different outputs in their RGB spaces. After training the deep CNN with abundant iterations, each filter from a color space represents a specific feature. Some groups of filters are smooth, such as Gabor filters. These Gabor-like filters may display from one to four different line structures. If these Gabor-like CNN filters are examined closely,

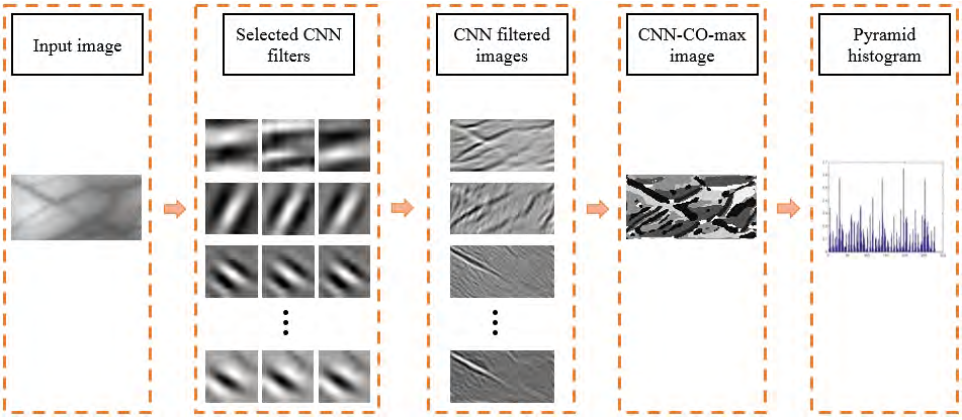

FIGURE 4. Framework of the proposed CNN-CO method for a finger vein image representation.

we find that they differ from Gabor filters in three respects. First, most of the learnable filters are not symmetrical. The asymmetrical structure shows different abilities for image representation. The differences from the CNN filtered images are shown in Fig. 3(b). Second, some of the filters show more complicated structures than Gabor filters. Third, most of the filters contain longer line structures, a characteristic that may capture the line structure features better. The CNN filtered images depicted in Fig. 3(b) demonstrate that the CNN filters have different capabilities in representing an image.

IV. PROPOSED CNN COMPETITIVE ORDER

As shown above, some learnable CNN filters have appearances and responses similar to Gabor filters, and the orientations of Gabor filters can be regarded as the orders of the CNN filters. Based on these observations and analysis, we adopt and select the CNN filters from a pre-trained CNN model. Specifically, a CNN-based local descriptor, named CNN Competitive Order (CNN-CO) is introduced for finger vein recognition. After the effective CNN filters are selected from the first layer of a well-trained CNN model, the proposed CNN-CO includes three steps for achieving an image representation: (1) computing the CNN filtered images, (2) building the competitive order (CO) image, and (3) generating the CNN-CO pyramid histogram. Fig. 4 describes the framework of the proposed CNN-CO method.

A. SELECTING THE CNN FILTERS

As shown in Fig. 3(a), CNN filters from different channels possess various levels of ability to describe local features. Some of the filters are smooth and display line structures, while some filters display noise and lack visual patterns. Hence, it is important to select the appropriate effective filters from the full set of CNN filters. In the following, we classify and select the CNN filters according to their appearances and outputs.

In light of the filters' appearances and responses, we define five types of labels for the CNN filters. The first label is *line*, which is used to describe the CNN filters with a line structure like a Gabor filter. Fig. 3(a) shows some examples of *line* filters, depicted with green boxes. It is observed that some of the filters labeled *line*, such as filters (1, 1), depict different line structures in RGB space. The orientations and widths of the filters (1, 1) from the separate R, G, and B space are different. Other *line* categories, such as the filters (5, 5) and filters (5, 6), show very similar structures in the filters' orientations and patterns in RGB space.

The second label, *smooth,* is used to represent the filters that have smooth distributions without regular line structures. Filters (1, 2) are examples of *smooth* filters that display a block distribution in the diagonal. In addition, the filters (1, 2) from RGB space show a similar distribution. Filters (1, 4) show a triangular diagonal distribution, which may have different ability to capture diagonal features in comparison with filters (1, 2). Although the filters labeled *smooth* do not contain regular line structures, the filtered images show similar image outputs as those *line* filters and Gabor filters.

The third label named *circle* represents the filters with circular structures. In Fig. 3(a), filters with this label are shown in blue boxes. The circular structure usually appears in the middle of the RGB filters and shows an inverse distribution outside the circle. Taking filters (5, 8) in Fig. 3 as an example, the first and third filters (from left to right) are depicted with a white circle and black background, while the second filter shows the inverse distribution. Although the first and third circles both show white color, their radii are different. When we look at the filtered images (5, 8), we see that the filter with a black circle and white background highlights vein structure. The images filtered by the filters that have a white circle and black background provide responses similar to the original image.

The fourth label, *noise*, represents the filters such as filters (5, 3) that lack any geometrical structure or distribution. In Fig. 3(a), these labels are shown in yellow boxes. The above four labels are mutually exclusive. Note, however, that some of the *line* and *smooth* filters show similar structures and distributions in three color spaces. To account for such cases, the label *similar* denotes the filters have a similar appearance and structure. *Similar* filters are highlighted with red boxes. Obviously, the *line* and *smooth* filters may have the label

TABLE 1. Group labels of CNN filters from the first layer of AlexNet network.

Features label	Order in the first layer
line	1,3,7,9,12,14,15,22,29,3436,37,38,44,49,50,51,53,54,57,59,60,61,64,65,66,69,70,71,72,73,74,75,76,78,80,81,82,87,88,89,91,92
smooth	2,4,5,6,13,17,18,19,20,21,24,26,27,28,30,31,42,46,47,48,58,62,63,67,68,83,84,85,86,90,94
similar	2,9,37,38,49,50,51,53,55,57,58,59,60,61,62,63,64,65,66,67,69,70,71,72,73,74,75,76,78,80,81,82,83,85,86,87,88,89,90,91,92,93
circle	10,16,25,32,39,40,41,45,52,56,79
noise	8,11,23,35,43,55,77,93,95,96

similar as well. Filters (1, 2) and (7, 1) are two examples of filters with two labels. Although the filters labeled *similar* have a similar appearance, tiny differences in appearance may be enlarged in their filtered images. Two groups of filtered images (5, 5) and (5, 6) show variances in global gray distribution and local vein representation in color spaces. Table 1 lists the labels of filters from the first layer of the AlexNet network according to their appearances and outputs.

As shown in Fig. 3(b), the image processed by the *line* and *smooth* filters show abundant orientation and local information. The *line* and *smooth* filters can be selected from the CNN filters for feature representation. The filters labeled *similar* are a subset of the combination of *line* and *smooth* filters. It is reasonable that these filters can be selected. As discussed above, the filtered images labeled *circle* always highlight the vein part. Since we aim at selecting Gabor-like filters that can capture line structure from a finger vein image, the *circle* filters are not selected. The filters labeled *noisy* are not well trained. Therefore, the *line*, *smooth*, and *similar* filters are selected in this work.

B. CONSTRUCTING AND USING THE CNN-CO

1) COMPUTING CNN FILTERED IMAGES

Since the deep CNN is trained using ImageNet, each group of CNN filters contains three filters from RGB spaces, which represent various features of an image. Furthermore, each filter from an RGB space has its own capability to represent an image. Suppose $C_m^c(x, y)$ and $CF_m^c(x, y)$ are the CNN filter and CNN filtered image from a color space at the serial number m, $m \in \{1, 2, \cdots, 96\}$. $c \in \{R, G, B\}$. $CF_m(x, y)$ represents the final filtered image using $I(x, y)$ as an input image. $CF_m^c(x, y)$ and $CF_m(x, y)$ can be obtained as:

$$CF_m^c(x, y) = C_m^c(x, y) * I(x, y), \quad (4)$$

$$CF_m(x, y) = CF_m^B(x, y) + CF_m^G(x, y) + CF_m^R(x, y), \quad (5)$$

where R, G, B represent the color spaces red, green, and blue, respectively.

2) BUILDING THE CO IMAGE

For a bank of selective CNN filtered images, the competitive order (CO) image and the competitive magnitude (CM) image with maximum rule can be achieved as follows:

$$CM_{\max}(x, y) = \max\left(CF_m(x, y)\right), \quad m = 1, 2, 3, \cdots, M, \quad (6)$$

$$CO_{\max}(x, y) = \arg\max_m \left(CF_m(x, y)\right), \quad m = 1, 2, 3, \cdots, M, \quad (7)$$

where M is the number of selected filters.

3) GENERATING THE CNN-CO PYRAMID HISTOGRAM

Different from the OC-based method that encodes a pixel with binary code, the proposed CNN-CO generates a histogram for an image description. Pyramid histogram description is a method frequently used to enhance the ability to discriminate features. It not only represents the local features of an image, but also extracts the spatial layout features from an image. For the pyramid histogram description, an input image is usually divided into a sequence of increasingly finer spatial grids by repeatedly increasing the number of divisions in each axis direction. The number of divisions can be selected as one or two. Then histograms are extracted from each grid, followed by concatenation to generate the final histogram. Given that l is the level order in a hierarchy and L is the pyramidal depth, the pyramid at level l has l^2 cells if the number of divisions is one.

For an input CNN-CO image, we build a local histogram for each grid, normalize each histogram using L_2 norm, and then concatenate each histogram to generate the final histogram for an image representation. Let H_t be a local histogram generated from a local patch, and ε is a very small constant. The L_2-norm scheme is defined as $H_t = \sqrt{\|H_t\|_2^2 + \varepsilon^2}$. The length of a CNN-CO histogram is

$$Length(CNN - CO) = M \times \sum_{l=1}^{L} l^2. \quad (8)$$

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

To verify the effectiveness of the proposed approach, the proposed CNN-CO approach is compared with the state-of-the-art local descriptors and OC-based methods in this section. Two public finger vein datasets MMCBNU_6000 [38], [39] and SDUMLA-HMT [40] are employed for our experiments. All the algorithms are computed using MATLAB (R2016b) on a computer with an Intel Core i7-5820K CPU and 16GB of RAM.

A. DATASETS AND EXPERIMENTS SETTINGS

In our previous work [38], the finger vein image dataset MMCBNU_6000 was established using a lab-made imaging device. MMCBNU_6000 consists of finger vein images captured from 100 volunteers, who are from 20 different countries. Each subject was asked to provide images of his/her index finger, middle finger, and ring finger of both hands. The collection of each finger was repeated 10 times to obtain 10 finger vein images. Thus, the finger vein database MMCBNU_6000 is composed of a total of 6,000 images.

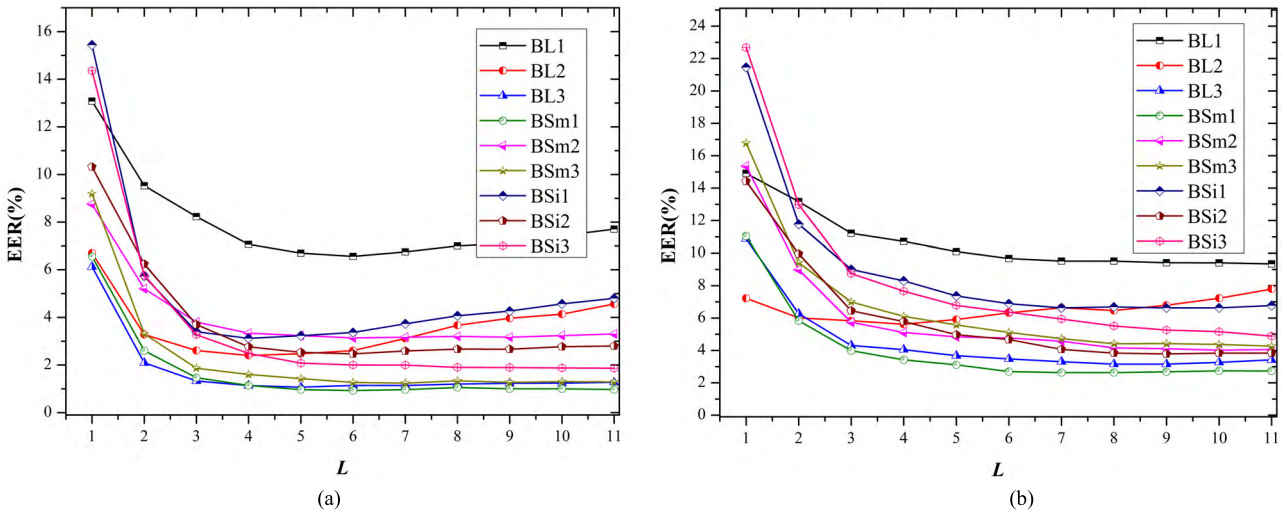

FIGURE 5. Comparison of *line*, *smooth*, and *similar* filters from different batches: (a) Results on MMCBNU_6000 and (b) Results on SDUMLA-HMT.

Each image was stored in BMP format with a resolution of 480 × 640 pixels. The captured images were localized to obtain regions of interest (ROI) using the flexible segmentation method proposed in [41]. The localized ROI images have a resolution of 60 × 128. The MMCBNU_6000 is available at [39].

SDUMLA-HMT is a multimodal biometric dataset [40], in which the finger vein sub-database is established from 106 individuals. In the capturing process, each subject was asked to provide images of his/her index finger, middle finger, and ring finger of both hands, and the collection for each finger is repeated for 6 times. Thus, the finger vein dataset is composed of 3,816 images. The image processing approach proposed in [42] was adopted to generate ROI images.

In this work, Euclidean distance is selected for computing the similarity between two pyramid histograms. The nearest neighbor classifier is employed for matching. The equal error rate (EER), which is the value where the false accept rate (FAR) is equal to the false reject rate (FRR), is adopted as the evaluation criterion for matching performance. A small EER indicates high matching performance. A match is accepted as genuine if two finger vein images are from the same finger; otherwise, there is no match, and one is regarded as an imposter. In all experiments, each finger is considered to be an individual. Half finger vein images from one individual are selected as the training set, while the remaining images are used as the test set. There are $3000(600 \times 5)$ genuine matches and $1,797,000(600 \times 599 \times 5)$ imposter matches for MMCBNU_6000. For SDUMLA-HMT, the numbers of genuine and imposter matches are $1,908(636 \times 3)$ and $1,211,580(636 \times 635 \times 3)$, respectively.

B. EXPERIMENTS ON THE SELECTED CNN FILTERS

The matching accuracy of the proposed approach depends on the selected CNN filters and the value of L in the pyramidal histogram description. In this subsection, four experiments are conducted to select the CNN filters and determine the optimal value of L.

1) COMPARING THE EFFECTIVENESS OF CNN FILTERS

The first experiment is designed to evaluate and compare the effectiveness of the *line*, *similar*, and *smooth* filters. As mentioned above, the *circle* filters cannot capture line-structure features. The *noise* filters usually are not well trained, which indicates they cannot extract effective features for image representation. Therefore, we compare only the effectiveness of the CNN filters labeled *line*, *smooth*, and *similar*.

The AlexNet network contains 96 groups of CNN filters from RGB space in the first layer. Usually, the number of channels in different layers and deep networks is 32. Without any other special purpose, we divide the CNN filters from the first layer into three batches according to their serial numbers. The number of selected CNN filters in each batch is no more than 32.

Fig. 5 compares the matching accuracy achieved using a single batch that contains either *line*, *smooth*, or *similar* filters only, with increasing value of L from 1 to 11. BL, BSm, and BSi represent the selected batches of filters labeled *line*, *smooth*, and *similar*, respectively. The numbers 1, 2, and 3 following BL, BSm, and BSi denote the order of the batch. Fig. 5 and Table 2 demonstrate that CNN filters from different batches show varied feature discrimination abilities, corresponding to different EER values. Among the nine batches, BL3 and BSm1 achieve competitive matching accuracy on two datasets. In addition, experimental results obtained on MMCBNU_6000 show that as the number of selected CNN filters increases from 1 to 6, the accuracy achieved by each single batch improves. However, by further increasing L, the accuracy become stable or worse. The same case is also achieved on SDUMLA-HMT. The inflection point appears when the value of L is increased from 7 to 9. The best accuracies are achieved by using the CNN filters

TABLE 2. Comparison of EER values using single batch and different combinations on MMCBNU_6000 and SDUMLA-HMT.

Combination	Number of selected filters in a batch	EER (%): MMCBNU_6000			EER (%): SDUMLA-HMT		
		L=4	L=5	L=6	L=7	L=8	L=9
BL1	9	7.07	6.69	6.56	9.52	9.51	9.41
BL2	15	2.40	2.47	2.60	6.63	6.48	6.80
BL3	19	1.13	1.07	1.14	3.31	3.15	3.15
BSm1	16	1.14	0.97	0.93	2.63	2.63	2.69
BSm2	7	3.34	3.24	3.14	4.58	4.15	4.10
BSm3	8	1.60	1.42	1.26	4.73	4.42	4.42
BSi1	2	3.12	3.23	3.37	6.63	6.69	6.62
BSi2	15	2.76	2.53	2.47	4.07	3.84	3.79
BSi3	25	2.48	2.08	2.00	5.94	5.52	5.26
BL1+BSm1	25	0.90	0.74	0.74	2.42	2.37	2.42
BL2+BSm2	22	3.10	3.10	3.10	4.53	4.36	4.37
BL3+BSm3	27	1.54	1.37	1.20	4.74	4.52	4.56
BL1+BL2+BL3	43	1.06	0.90	0.93	2.98	2.95	2.99
BSm1+BSm2+BSm3	31	1.38	1.23	1.11	3.68	3.47	3.47
BSi1+BSi2+BSi3	42	1.94	1.70	1.50	4.73	4.37	4.37

from the batch BSm1 on two datasets, and the EERs are 0.93% ($L = 5$) and 2.62% ($L = 7$) on MMCBNU_6000 and SDUMLA-HMT. This demonstrates that the optimal CNN filters from a single batch (BSm1) is robust for different datasets.

In comparison, although batch BSi1 contains only two groups of selected CNN filters, the matching accuracy achieved using BSi1 is better than that achieved using BL1 (with nine groups of CNN filters). In addition, BL2, BSm1 and BSi2 have almost same number of selected CNN filters, however, BSm1 achieves much better results than BSm1 and BSi2 in all kinds of pyramidal methods on two datasets.

2) COMPARING COMBINATION OF CNN FILTERS

As analyzed above, the single batches from each label yield different matching accuracies. In the second experiment, we investigate the effectiveness of combinations of the above batches on two datasets. The values of L are selected as the regions of inflection points on two datasets in this subsection. Note that there are many possible combinations of nine batches. We focus on two combinations only. The first combines the selected CNN filters from the same order of a batch, while the second combine the selected CNN filters with the same labels regardless of batch. For each combination, two or three groups of CNN filters from the batches are concatenated to generate a large batch. For example, the combination BL1 and BSm1 have 25 groups of CNN filters. Then the same framework shown in Fig. 4 is used to process the 25 combined CNN filters to get the pyramid features.

Table 2 shows the matching accuracies achieved using all six combinations. Since the filters labeled *similar* are a subset of the combination of filters labeled *line* and *smooth*, only the CNN filters with labels *line* and *smooth* are explored for the combination method using the same batch order. The combination BL1+BSm1 produces better matching performance than using single batch BL1 or BSm1 alone on two datasets. The best accuracy for BL1+BSm1 is achieved by using $L = 5$ and $L = 8$, and the EER values are 0.74% and 2.37% on MMCBNU_6000 and SDUMLA-HMT, which are 25.8% and 9.89% improvements over BSm1 alone. It is noted that the combination of BL2+BSm2 or BL3+BSm3 reduces matching performance.

While combining CNN filters with the same label but from different batch orders, we find that combining just the batches labeled *line*r can achieve better accuracy than any single batch on MMCBNU_6000. However, the combination of three batches with the label *smooth* and *similar* from different orders provides poorer performance than using a single batch. For the results on SDUMLA-HMT, the combinations of same label achieve poor accuracies than using a single batch for BL, BSm, and BSi.

In addition, although the combination BL1+BSm1 contains a smaller number of selected CNN filters than four other combinations, it shows much better matching accuracy. Furthermore, the combination BL1+BSm1 is also efficient in feature extraction and matching since it contains few selected filters. It is concluded that the categories of the selected CNN filters are much more important than the number of selective CNN filters.

3) SEARCHING FOR THE OPTIMAL VALUE OF L

The depth of the pyramidal extraction of histograms can also affect the matching accuracy and speed. Tables 2 compares the accuracy achieved using a single batch and different combinations with various values of L on two datasets. We can see clearly that a large L yields high matching accuracy from $L = 1$ to the inflection points. While further increasing L from the inflection points, the matching accuracies become stable and worse. The selected filters from a batch or a combination achieve the best or competitive accuracies when $L = 5$ and $L = 8$ on MMCBNU_6000 and SDUMLA-HMT, respectively. Among all the single batches and combinations, the best accuracy is achieved on the combination BL1+BSm1 using $L = 5$ and $L = 8$, and the EER values are 0.74% and 2.37%. On the other hand, as the Formula (8) shows, the large L yield large length of histogram, which bring long time in feature extraction and matching. Integrated into account the matching accuracy and processing time, the optimal values of L are selected as 5 and 8 for MMCBNU_6000 and SDUMLA-HMT.

TABLE 3. Comparison of the proposed CNN-CO using selected filters with different color spaces on MMCBNU_6000 and SDUMLA-HMT.

Selected filters	EER (%) using L=5: MMCBNU_6000					EER (%) using L=8: SDUMLA-HMT				
	R	G	B	RGB	Feature concatenation	R	G	B	RGB	Feature concatenation
BL3	1.40	1.63	1.60	1.07	1.13	3.16	3.30	5.43	3.15	3.23
BSm1	1.71	26.16	1.50	0.97	1.33	4.99	28.59	4.10	2.63	4.46
BL1+BSm1	1.70	26.18	1.50	0.74	1.31	4.73	28.59	4.09	2.37	4.32
BL1+BL2+BL3	1.63	15.53	42.72	0.90	2.12	16.09	41.52	4.73	4.37	5.31

TABLE 4. Comparison of the proposed CNN-CO with state-of-the-art algorithms.

Algorithm	EER (%)		Processing time (ms): MMCBNU_6000	
	MMCBNU_6000	SDUMLA-HMT	Feature extraction time	Matching time
LBP	2.40	8.21	39.19	93.67
HOG	1.04	3.23	3.53	0.31
Gabor filter	2.66	10.56	22.98	0.44
Steerable filter	3.47	5.98	7.51	0.13
LDC	3.29	8.03	**2.11**	10.12
MDC	1.46	3.89	30.81	21.24
CompCode	2.40	4.94	54.74	31.41
HCO-33	**0.53**	**1.53**	36.36	0.11
HCO-11	2.57	9.47	9.68	**0.09**
Proposed CNN-CO	0.74	2.37	39.62	0.64

4) COMPARING THE FILTERS FROM COLOR SPACES

Convolutional filters from the CNN model pre-trained on ImageNet are all from RGB spaces. Although the summation of CNN filtered images from RGB spaces is used to build the CO, we can also use the filter from one color space to construct the proposed CNN-CO. In this part of our work, we study the effect of filters from different color spaces on the proposed CNN-CO method. Since the Formula (5) can be viewed as a kind of feature fusion strategy, we also compare the RGB summation strategy with feature-level fusion.

BL3, BSm1, BL1+BSm1, and BL1+BL2+BL3 are selected for comparison in color spaces since they all show competitive performance. Table 3 lists the performances of the selected filters in different color spaces on two datasets. It is seen from Table 3, the proposed CNN-CO using RGB spaces achieves the best performance when compared with the results using CNN-CO features from B, G, and R spaces separately. The EER calculated using the selected filters from the G space of BSM1 is 26.16% on MMCBNU_6000, which is much higher than using the other filters and color spaces. Other higher EER values are 15.53% and 42.72%. We ascribe the higher EER values of some selected filters in a color space to the idea that these filters have an effect similar to *circle* filters. However, when the final filtered image is achieved by adding filtered images from the RGB spaces, better features are obtained. In addition, the fused filtered image contains more abundant information. Hence, the CNN-CO features extracted from the fused image can achieve better accuracy. For example, the EER of CNN-CO features extracted from BSm1 using RGB is 0.97% on MMCBNU_600, which is a much better result than using a single color space. Moreover, it is found that different batches or combinations from various color spaces achieve different performances. For example, BL3 obtains the smallest EER from R space, while BSm1 achieves the best accuracy from B space on two datasets.

We also compare the proposed approach with the frequently used feature fusion that concatenates the CNN-CO features from three spaces. Experimental results on two datasets show that the proposed approach by fusing features from RGB spaces can achieve better matching accuracy than that concatenate features from RGB space.

C. COMPARING OTHER DESCRIPTORS

To validate our proposed approach further, we compare the proposed CNN-based local descriptor CNN-CO with nine commonly-used local descriptors: LBP [28], histogram of oriented gradients (HOG) [43], Gabor filter [42], Steerable filter [44], and the state-of-the-art approaches on line features including local directional code (LDC) [45], minimum directional code (MDC) [46], competitive code (CompCode) [19], and two histograms of competitive orientation (HCO-33 and HCO-11) [3] as show in Table 4 and Fig. 6.

Fig. 6 show the performance comparison in terms of ROC curves on two public datasets. HCO used Gabor filters with kernel sizes 33 and 11, respectively. We can see that the proposed CNN-CO performs better than most of the compared algorithms on two public finger vein datasets. These results illustrate the competitive features extracted using the selected CNN filters are effective. The HCO using Gabor filters with kernel size 33 performs best among all the approaches. However, when the kernel size is reduced to the same as the AlexNet filters, the accuracy achieved using HCO-11 degrades considerably, with the EER increasing from 0.53% to 2.57% on MMCBNU_6000 and from 1.53% to 9.47% on SDUMLA-HMT. These results reveal that the kernel size of a filter affects the matching accuracy when using the competitive features. With a larger filter size in a deep network, the proposed CNN-CO may achieve better accuracy.

In this section, we also compare the average feature extraction time and matching time of all the approaches in Table 4. The MMCBNU_6000 is selected for processing

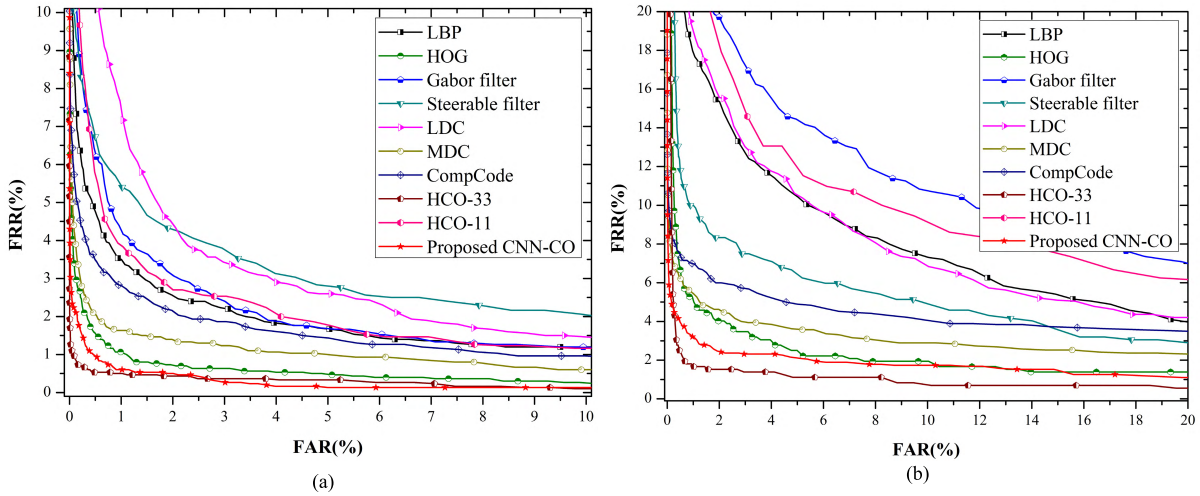

FIGURE 6. Comparison of the proposed CNN-CO with state-of-the-art algorithms. (a) Results on MMCBNU_6000 and (b) results on SDUMLA-HMT.

time comparison. Euclidean distance is employed to compute similarity for HOG, Gabor filter, Steerable filter, HCO-33 and HCO-11, while Hamming distance is selected for LBP, LDC, MDC, and CompCode. As shown in Table 4, binary code features usually take more processing time in feature extraction and matching. The histogram-based local descriptors such as HOG, Steerable filter, and HCO take less time for feature extraction. Due to small feature size and the efficiency of Euclidean distance, the matching time for histogram-based approaches is negligible. Compare with the approaches using Gabor filters such as MDC, CompCode and HCO, the proposed method extracts feature from RGB color spaces. It hence spends more time for feature extraction. However, the feature extraction time using the propose CNN-CO is only 39.62ms for an image, which is fast enough for real-time applications.

VI. CONCLUSION

Deep convolution neural networks have been used widely in many image classification tasks and have achieved promising accuracy when a large-scale labelled dataset is available. For finger vein recognition, however, the number and size of finger vein datasets available to train a CNN model is limited. In this paper, the problem is addressed by selecting CNN filters from a pre-trained CNN model, by which, a CNN-based local descriptor named CNN-CO is proposed. By visualizing and analyzing the CNN filters from the first layer of the AlexNet network, along with their corresponding filtered images, we classify the CNN filters using five labels based on the appearances and outputs in RGB spaces. The selected CNN filters are employed to generate the competitive order (CO) image using the maximum rule. Then we concatenate the pyramidal histograms from different levels as calculated from the competitive order image to build the final histogram. Experimental results demonstrate that the selected CNN filters from the first layer of AlexNet are effective for finger vein image representation and the recognition performance by the proposed CNN-CO outperforms the state-of-the-art methods.

In this research, the CNN filters employed in the proposed CNN-CO are selected according to the appearance and responses of the filters. In future, we will design a better approach to extract effective CNN filters automatically from a convolution layer, thereby allowing the proposed CNN-CO to be extended to use other deep CNN models. In addition, we believe that the proposed CNN-CO can be used for other biometrics that have abundant line structures, e.g., palm print recognition.

REFERENCES

[1] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two-stream convolutional network learning," *Neurocomputing*, vol. 290, pp. 100–107, May 2018.
[2] L. Yang, G. Yang, X. Xi, X. Meng, C. Zhang, and Y. Yin, "Tri-branch vein structure assisted finger vein recognition," *IEEE Access*, vol. 5, pp. 21020–21028, 2017.
[3] Y. Lu, S. Wu, Z. Fang, N. Xiong, S. Yoon, and D. S. Park, "Exploring finger vein based personal authentication for secure IoT," *Future Generat. Comput. Syst.*, vol. 77, pp. 149–160, Dec. 2017.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[5] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[8] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3119–3127.
[9] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.
[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[11] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[12] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[13] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016, pp. 20–36.
[14] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2014, pp. 392–407.
[15] J. Wang, Z. Pan, G. Wang, M. Li, and Y. Li, "Spatial pyramid pooling of selective convolutional features for vein recognition," *IEEE Access*, vol. 6, pp. 28563–28572, 2018.
[16] Y. Zhou and A. Kumar, "Human identification using palm-vein images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1259–1274, Dec. 2011.
[17] W. Jia, D.-S. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognit.*, vol. 41, no. 5, pp. 1504–1513, 2008.
[18] W. Zuo, F. Yue, and D. Zhang, "On accurate orientation extraction and appropriate distance measure for low-resolution palmprint recognition," *Pattern Recognit.*, vol. 44, no. 4, pp. 964–972, 2011.
[19] A. W.-K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Aug. 2004, pp. 520–523.
[20] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.
[21] Y. Lu, S. Yoon, S. J. Xie, J. Yang, Z. Wang, and D. S. Park, "Finger vein recognition using histogram of competitive Gabor responses," in *Proc. Int. Conf. Pattern Recognit.*, Stockholm, Sweden, Aug. 2014, pp. 1758–1763.
[22] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, no. 4, pp. 194–203, 2004.
[23] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vol. E90-D, no. 8, pp. 1185–1194, 2007.
[24] W. Song, T. Kim, H. Kim, J. Choi, H. Kong, and S. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, 2011.
[25] J.-D. Wu and C.-T. Liu, "Finger-vein pattern identification using principal component analysis and the neural network technique," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5423–5427, 2011.
[26] J.-D. Wu and C.-T. Liu, "Finger-vein pattern identification using SVM and neural network technique," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 14284–14289, 2011.
[27] C. Liu and Y.-H. Kim, "An efficient finger-vein extraction algorithm based on random forest regression with efficient local binary patterns," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3141–3145.
[28] E. C. Lee, H. C. Lee, and K. R. Park, "Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction," *Int. J. Imag. Syst. Technol.*, vol. 19, no. 3, pp. 179–186, 2009.
[29] B. A. Rosdi, W. S. Chai, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, 2011.
[30] Y. Lu, S. Yoon, S. J. Xie, J. Yang, Z. Wang, and D. S. Park, "Efficient descriptor of histogram of salient edge orientation map for finger vein recognition," *Appl. Opt.*, vol. 53, no. 20, pp. 4585–4593, 2014.
[31] H. Houjun, S. Liu, H. Zheng, L. Ni, Z. Yi, and W. Li, "DeepVein: Novel finger vein verification methods based on deep convolutional neural networks," in *Proc. IEEE Int. Conf. Identity, Secur. Behav. Anal. (ISBA)*, Feb. 2017, pp. 1–8.
[32] H. G. Hong, M. B. Lee, and K. R. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, p. 1297, 2017.
[33] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Pattern Recognit. Lett.*, vol. 119, pp. 148–156, Mar. 2019.
[34] H. Qin and M. A. El-Yacoubi, "Deep representation for finger-vein image-quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1677–1693, Aug. 2018.
[35] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint representation for personal identification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2005, pp. 279–284.
[36] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.
[37] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2014, pp. 818–833.
[38] Y. Lu, S. J. Xie, S. Yoon, Z. Wang, and D. S. Park, "An available database for the research of finger vein recognition," presented at the 6th Int. Congr. Image Signal Process., Hangzhou, China, Dec. 2013.
[39] *MMCBNU_6000*. Accessed: Sep. 2018. [Online]. Available: http://multilab.jbnu.ac.kr/MMCBNU_6000
[40] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Biometric Recognition* (Lecture Notes in Computer Science), vol. 7098, Z. Sun, J. Lai, X. Chen, and T. Tan, Eds. Berlin, Germany: Springer, 2011, pp. 260–268.
[41] Y. Lu, S. J. Xie, S. Yoon, J. Yang, and D. S. Park, "Robust finger vein ROI localization based on flexible segmentation," *Sensors*, vol. 13, no. 11, pp. 14339–14366, 2013.
[42] S. J. Xie, J. Yang, S. Yoon, Y. Lu, and D. S. Park, "Guided Gabor filter for finger vein pattern extraction," in *Proc. 8th Int. Conf. Signal Image Technol. Internet Based Syst. (SITIS)*, Nov. 2012, pp. 118–123.
[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.
[44] J. Yang and L. Xu, "Efficient finger vein localization and Recognition," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2010, pp. 1148–1151.
[45] X. Meng, G. Yang, Y. Yin, and R. Xiao, "Finger vein recognition based on local directional code," *Sensors*, vol. 12, no. 11, pp. 14937–14952, 2012.
[46] K.-S. Wu, J.-C. Lee, T.-M. Lo, K.-C. Chang, and C.-P. Chang, "A secure palm vein recognition system," *J. Syst. Softw.*, vol. 86, no. 11, pp. 2870–2876, 2013.

YU LU received the M.S. degree from the Jiangxi University of Finance and Economics, Nanchang, China, in 2011, and the Ph.D. degree from Chonbuk National University, Jeonju, South Korea, in 2014. He is currently a Researcher with SenseTime Research. His research interests include image processing, biometrics, pattern recognition, object detection, image classification, and machine learning.

SHANJUAN XIE received the M.S. and Ph.D. degrees from Chonbuk National University, South Korea, in 2009 and 2012, respectively. She has authored almost 40 papers in his research related international journals and conferences. She is currently an Associate Professor with Hangzhou Normal University. Her areas of interest include biometrics, pattern recognition, image processing, and machine learning.

SHIQIAN WU (M'02–SM'05) received the B.Eng. and M.Eng. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1985 and 1988, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2001. He was an Assistant Professor, Lecturer, and then an Associate Professor with HUST, from 1988 to 1997. From 2000 to 2014, he was a Research Fellow then a Research Scientist with the Agency for Science, Technology, and Research, Singapore. He is currently a Professor with the School of Machinery and Automation, Wuhan University of Science and Technology, Wuhan. He has co-authored the book *Dynamic Fuzzy Neural Networks* (Singapore: McGraw-Hill, 2003). He has authored or co-authored over 190 scientific publications (book chapters and journal/conference papers). He was listed as one of the Most Cited Chinese Researchers by Elsevier, in 2017. His current research interests include image processing, pattern recognition, machine vision, fuzzy systems, and neural networks.